

# ICES REPORT 12-38

---

August 2012

## Blended Isogeometric Shells

by

D.J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes



**The Institute for Computational Engineering and Sciences**  
The University of Texas at Austin  
Austin, Texas 78712

*Reference: D.J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes, Blended Isogeometric Shells, ICES REPORT 12-38, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, August 2012.*

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>AUG 2012</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2012 to 00-00-2012</b>	
4. TITLE AND SUBTITLE <b>Blended Isogeometric Shells</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of Texas at Austin, The Institute for Computational Engineering and Sciences, Austin, TX, 78712</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <b>We propose a new isogeometric shell formulation that blends Kirchhoff-Love theory with Reissner-Mindlin theory. This enables us to reduce the size of equation systems by eliminating rotational degrees of freedom while simultaneously providing a general and effective treatment of kinematic constraints engendered by shell intersections, folds, boundary conditions the merging of NURBS patches, etc. We illustrate the blended theory's performance on a series of test problems.</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>34</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

# Blended Isogeometric Shells

D.J. Benson<sup>a,1</sup>, S. Hartmann<sup>b,2</sup>, Y. Bazilevs<sup>a,3</sup>, M.-C. Hsu<sup>a,4</sup>, and  
T.J.R. Hughes<sup>c,5</sup>

<sup>a</sup>*Department of Structural Engineering, University of California, San Diego 9500 Gilman Drive, La Jolla, CA 92093, USA*

<sup>b</sup>*DYNAmore GmbH, Industriestr. 2, 70565 Stuttgart-Vaihingen, Germany*

<sup>c</sup>*Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 East 24th Street, 1 University Station C0200, Austin, TX 78712, USA*

---

## Abstract

We propose a new isogeometric shell formulation that blends Kirchhoff-Love theory with Reissner-Mindlin theory. This enables us to reduce the size of equation systems by eliminating rotational degrees of freedom while simultaneously providing a general and effective treatment of kinematic constraints engendered by shell intersections, folds, boundary conditions, the merging of NURBS patches, etc. We illustrate the blended theory's performance on a series of test problems.

*Key words:* isogeometric analysis, NURBS, shells, rotation-free, nonlinear

---

## 1 Introduction

Reissner-Mindlin shell theory, also referred to as “thick shell theory,” which accommodates transverse shear deformations, has become the predominate theory used as a basis of finite element implementations. The main attribute of Reissner-Mindlin theory, as far as finite element technology goes, is that it is  $C^0$ -conforming, that is, standard  $C^0$ -continuous interpolation functions are appropriate for representing displacements and independent fiber rotations. Various “locking” phenomena are

---

<sup>1</sup> Professor and corresponding author. Phone: 858-534-5928. E-mail: dbenson@ucsd.edu

<sup>2</sup> Research Engineer

<sup>3</sup> Associate Professor

<sup>4</sup> Graduate Research Assistant

<sup>5</sup> Professor of Aerospace Engineering and Engineering Mechanics, Computational and Applied Mathematics Chair III

a hindrance but are either overcome or mitigated sufficiently by technologies that were initially developed in the 1970s and refined by numerous researchers subsequently. The basic four-node bilinear Reissner-Mindlin element, appropriately treated, has proved particularly efficient and robust, and is the workhorse of commercial finite element programs, such as LS-DYNA [14], which is extensively utilized for automobile crash dynamics and sheet metal forming.

A particular strength in this context is the generality of the formulation in that it is applicable to arbitrarily large deformations and fully nonlinear elastic and inelastic constitutive behavior. It is fair to say that Reissner-Mindlin elements have almost completely dominated finite element shell analysis for over 30 years.

Historically, initial efforts to develop shell elements were based on Kirchhoff-Love theory, also known as “thin shell theory.” Kirchhoff-Love theory invokes the Kirchhoff constraint of zero transverse shear strains, and requires  $C^1$ -continuous finite element basis functions for the transverse displacements. Some functions of this type exist, but these lead to extremely complex elements that are difficult to use in a general nonlinear computing environment. Ultimately, these elements could not compete with the far simpler, more efficient and robust Reissner-Mindlin elements. Within the traditional finite element paradigm,  $C^1$ -continuous basis functions require *derivative* degrees of freedom.

With the advent of isogeometric analysis [19,12] a plethora of  $C^1$ -continuous basis functions is available and new ones are under development (see e.g. [11,27,25]). These offer unique finite element possibilities in that their smoothness properties do not involve derivative degrees of freedom. Consequently, so-called “rotation-free” elements become a possibility.

This idea was first proposed in Hughes et al. [19] and has been developed by Kiendl et al. [21,20] and Benson et al. [6]. As the name implies, rotation-free Kirchhoff-Love elements do not involve rotational degrees of freedom. This represents a reduction in the total number of degrees of freedom and also the bandwidth size, by roughly a factor of two each. In implicit analysis, compared with the use of Reissner-Mindlin elements, storage of the left-hand-side jacobian matrix will be reduced by approximately a factor of four, and factorization cost will be reduced by approximately a factor of eight. These savings are very significant and potentially could represent a decisive advantage when compared to the use of Reissner-Mindlin elements. In explicit analysis, storage would be reduced somewhat, but not by a significant enough amount to be advantageous, and compute cost might be commensurate, but slightly greater than Reissner-Mindlin elements, due to the necessity of computing second-derivatives, or equivalents (see Benson et al. [6] for a variant not involving directly computing second derivatives). Nevertheless, the potential advantage in implicit analysis is so significant that it warrants further investigation. Unlike the case for Reissner-Mindlin elements, areas in the shell that are only geometrically  $C^0$ -continuous require special treatment. Physically, these are

associated with sharp folds, creases and stiffener junctions. Along these  $C^0$  lines, the angle between intersecting branches needs to be maintained during deformation. Various ways of introducing these constraints have been investigated. Kiendl et al. [20,21] have investigated different implementations of linear constraint equations and penalized “bending strips.” There are other possibilities that heretofore have not been investigated, namely, “rigid bodies” [7]. Another need for the use of these techniques is in multi-patch NURBS meshes of even smooth shells in which basis functions at the patch boundaries only possess  $C^0$ -continuity.

It is our opinion that none of these constraint techniques is completely satisfactory in all cases. Even if they can be used successfully in certain situations they can give rise to reduced critical time steps in explicit dynamic analysis and preclude plastic hinge formation at shell intersections and at boundaries, where often there are very large bending moments. The plastic hinges are precluded by the fact that there are no transverse shear strains in rotation-free elements. As a result, in situations such as these Reissner-Mindlin elements are much more effective, but these of course require rotational degree of freedom. Our conclusion is that the potential advantages of rotation-free elements in implicit analysis are presently difficult to realize in most engineering situations.

In this work we offer what we believe is a novel and practical solution to these difficulties. It is based on the concept of a “blended shell theory” in which the assumptions underlying classical Reissner-Mindlin and Kirchhoff-Love theories are combined. Specifically, we describe a blended shell through a linear combination of the kinematical assumptions of Reissner-Mindlin and Kirchhoff-Love theories. We recall that the only difference between the two theories involves the description of the kinematics of the fiber vectors. In the case of Kirchhoff-Love theory, the fiber vector remains orthogonal to the lamina reference surface throughout deformation (see Hughes and Liu [17] for a description of the concepts of shell laminae and fibers), whereas in Reissner-Mindlin theory it depends on the independent rotation fields.

The parameter defining the linear combination is a function of the coordinates, but it is convenient to implement it discretely in terms of control variables (i.e., “nodes”). Our blended theory can represent both extremes, that is, an entire Reissner-Mindlin shell, or an entire Kirchhoff-Love shell. It can also represent a judicious combination of the two to deal with the issue of constraints. In this case, we propose to select all the control variables where there are intersections, boundaries, folds, or  $C^0$  interfaces as Reissner-Mindlin control variables, and elsewhere in the shell as Kirchhoff-Love variables. The result provides for the elimination of rotational degrees of freedom in the smooth regions of the shell and the robust treatment of intersections, boundaries, folds, etc., provided by Reissner-Mindlin theory precisely where it is needed. We believe this represents an ideal solution that may provide optimally efficient shell discretizations for a wide range of practical problems. An application in sheet metal forming is schematically illustrated in Figure 1.

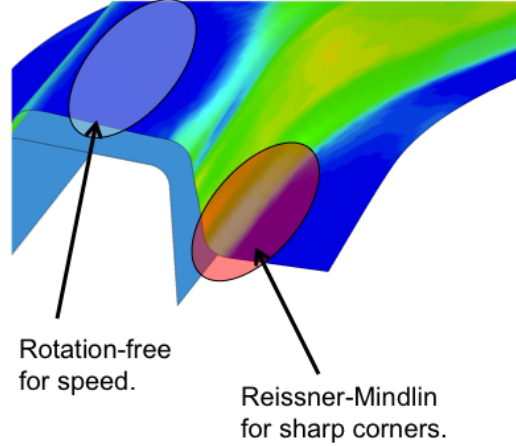


Fig. 1. Schematic illustration of an application of the blended shell theory for an implicit simulation of metal forming. Regions with low curvature may be efficiently analyzed with rotation-free thin shells while regions of high curvature would use the Reissner-Mindlin formulation.

The idea of a blended shell theory can go even further. Specifically, higher-order shell theories involve more complex descriptions of fiber behavior. These can be blended with Kirchhoff-Love and/or Reissner-Mindlin theories to achieve efficient formulations for composite laminates and sandwich shells, and they may also be useful in biomechanical modeling (e.g. arteries). It is also possible to go still further with the concept and blend shell theories with continuum solid theories in the spirit of transition elements. Again biomechanical modeling opportunities present themselves, such as for heart-artery models. We also note that all these blended theories can be developed within the IGA format of exact CAD modeling.

The blended formulation presented here is valid for a broad class of basis functions with  $C^1$  or greater continuity, but in our examples it is applied to B-splines and NURBS. Their form is summarized in Section 2. This is followed in Section 3 by a brief summary of current methods for joining patches in rotation-free shell analysis. A detailed presentation of how the blended formulation may be efficiently implemented is provided in Section 4. Linear and nonlinear example calculations follow in Section 5 to show the strengths and weaknesses of the various approaches to multi-patch analysis, and the good behavior of the blended formulation in all cases.

## 2 A summary of B-spline and NURBS basis functions

The literature on the basis functions used in CAD and CG is extensive. The standard reference for NURBS is by Piegl and Tiller [23], and a popular introductory book is by Rogers [24]. The first paper on isogeometric analysis [19] contains a concise introduction to B-splines and NURBS, and the definitive treatment of isogeometric

analysis is provided by Cottrell, Hughes and Bazilevs [12]. A brief summary is provided here to establish the notation used in the remainder of the paper.

A B-spline curve is defined by a *knot vector* and a vector of *control points*. The knot vector defines the set of basis functions  $N_A^b(s)$  in terms of the parametric coordinate  $s$  through the recursive relation

$$N_{A,0}^b(s) = \begin{cases} 1 & \text{if } s_A \leq s < s_{A+1} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

$$N_{A,p}^b(s) = \frac{s - s_A}{s_{A+p} - s_A} N_{A,p-1}^b(s) + \frac{s_{A+p+1} - s}{s_{A+p+1} - s_{A+1}} N_{A+1,p-1}^b(s) \quad (2)$$

where  $p > 0$  is the degree of the B-spline. For brevity, the degree is omitted in the remainder of the paper.

The knot vectors used here are all open. The first and last knots have multiplicity  $p+1$  for a B-spline of polynomial degree  $p$ . Knots may be repeated in the interior of the knot vector, with each repetition locally lowering the degree of continuity by one. The locations of the knots define the boundaries of the elements in the parametric space. Note that the basis functions are in general not interpolatory except at the boundaries, but they do satisfy the partition of unity property,  $\sum_A N_A(s) = 1$ , and are non-negative everywhere.

A basis function of degree  $p$  spans up to  $p+1$  elements. Starting from the left boundary, the first basis function spans one element, the second spans two elements and so on until the basis functions span  $p+1$  elements. The same progression of spanning from 1 to  $p+1$  elements occurs working to the left from the right boundary.

The control point  $\mathbf{x}_A$  is associated with the basis function  $N_A^b$  to define the curve

$$\mathbf{x}(s) = \sum_A N_A^b(s) \mathbf{x}_A. \quad (3)$$

Note that the control points do not necessarily lie on the curve they define, except for the first and last ones.

Surfaces are defined by a Cartesian product of one-dimensional B-spline curves,

$$\mathbf{x}(s_1, s_2) = \sum_{AB} N_{AB}^b(s_1, s_2) \mathbf{x}_{AB} \quad \text{where} \quad N_{AB}^b(s_1, s_2) = N_A^b(s_1) N_B^b(s_2). \quad (4)$$

For simplicity, the subscript  $AB$  is replaced by a single subscript in the remainder of the paper. As a result of their construction, the two-dimensional basis functions are in general not interpolatory.

A NURBS curve is defined by introducing a fourth, homogeneous coordinate  $w > 0$  at each control point,

$$\mathbf{x}(s) = \sum_A \left( \frac{N_A^b(s)w_A}{\sum_B N_B^b(s)w_B} \right) \mathbf{x}_A. \quad (5)$$

The fourth coordinate is not an unknown, but specified with the initial coordinates  $\mathbf{x}_A$  to define the reference configuration, and it remains constant during the analysis. By inspection, the NURBS basis functions  $N_A$  are

$$N_A(s) = \frac{N_A^b(s)w_A}{\sum_B N_B^b(s)w_B}. \quad (6)$$

NURBS surfaces are again defined by a Cartesian product of the one-dimensional basis functions, and, again, the double subscript is replaced by a single subscript,

$$\mathbf{x}(s_1, s_2) = \sum_A N_A(s_1, s_2) \mathbf{x}_A. \quad (7)$$

NURBS curves and surfaces have the same properties as B-spline curves and surfaces:

- (1) They form a partition of unity,  $\sum_A N_A = 1$ .
- (2) The support of each  $N_A$  is compact, spanning up to  $p + 1$  knot intervals (elements).
- (3) The basis functions are non-negative.  $N_A \geq 0$ .
- (4) The basis functions are not usually interpolatory.
- (5) The interior of the patch may be continuous up to  $C^{p-1}$  but the continuity between patches is  $C^0$  unless constraints are imposed to increase it.
- (6) The coordinates of the control points do not necessarily lie on the surface.

### 3 Multi-patch analysis

NURBS and B-spline patches are naturally assembled in a manner similar to traditional finite elements to build domains that are more topologically complicated than a logically rectangular mesh. While NURBS and B-splines can be up to  $C^{p-1}$  on their interiors, they meet at  $C^0$  boundaries. For the analysis of solids, or for shear deformable structural formulations, this does not pose any problem because their formulation requires only  $C^0$  continuity. Thin shell formulations, on the other hand, require  $C^1$  continuity to transmit bending moments, a condition which is violated at the patch boundaries. The  $C^0$  boundaries between patches behave like piano hinges for thin shell formulations, and therefore  $C^1$  continuity must be enforced along the shared boundary to transmit the moments.



Joining patches edge-to-edge to form a single  $C^1$  surface is the simplest type of  $C^0$  merge occurring in structural analysis. Much more common are the cases where the final structure retains the  $C^0$  lines, for example, as in the square tube buckling problem shown in Section 5.4. In the case of the square tube, there are sharp  $90^\circ$  corners that must be maintained, making the  $C^0$  edge line a genuine geometric feature. A simple T-intersection is another case of a  $C^0$  line that is also frequently encountered in real structures. Like a sharp corner, the low level of continuity is an intrinsic property of the exact geometry that cannot be eliminated by a different choice of basis functions. The problem for merging NURBS patches that define a smooth surface can be addressed by using different basis functions, e.g., T-Splines [27], but the problems with sharp edges and corners remain.

There are a number of different ways to address these issues. For example, the rotation-free formulation by Oñate and Flores [22] uses  $C^0$  basis functions in a novel way. Within the context of isogeometric analysis, linear constraints formulated in a master-slave form [21] have been used. The linearity stems from the assumption that the membrane deformations and rigid body rotations are small, conditions that are not satisfied in general for nonlinear problems.

Rigid bodies [7] appear to be the natural generalization of linear constraints to the nonlinear regime, offering a symmetry in their treatment of adjacent patches that has been found useful in contact. The coordinates of all points in a rigid body are constant in the body's local reference frame. If the points satisfy a linear constraint in the local reference frame at  $t = 0$ , then the constraint will be satisfied at all later times in the local frame independent of any large rigid body rotations. Instead of spending a lot of time deriving potentially complicated linear constraints between adjacent patches (e.g., for the L-shaped domain used in an Section 5.3), it is sufficient to simply group each independent set of control points into a rigid body. For example, if two patches share ten control points along a common boundary, there are ten sets of three linearized master-slave constraints, each having three control points (the shared control point plus an additional one from each patch). The ten sets are used to define ten rigid bodies, each consisting of the appropriate set of three control points. The particular rigid body formulation used in the example calculations is summarized in the Appendix.

Care must be exercised in introducing constraints. As shown in Section 5.2, even simple eigenvalue problems may have significant sensitivity to the choice of constraint equations. Generating the correct constraints for general structural topologies and basis functions in an automated manner is anticipated to be a difficult problem.

Another approach is the penalty method, e.g., as embodied in the bending strip method [2]. It is assumed that the shell structure is comprised of smooth subdomains, such as NURBS patches, that are joined with  $C^0$ -continuity. In addition, thin strips of fictitious material, also modeled as surface NURBS patches, are placed at

structural patch intersections. The triples of control points at the patch interface, consisting of a shared control point and one on each side, are extracted and used as a control mesh for the bending strips. The parametric domain of each bending strip consists of one quadratic element in the direction transverse to the interface and, for simplicity and computational efficiency, as many linear elements as necessary to accommodate all the control points along the length of the strip. The material is assumed to have zero mass, zero membrane stiffness, and non-zero bending stiffness only in the direction transverse to the interface. Bending strips transmit the bending moment from one patch to the other, which may not be accomplished with a  $C^0$ -continuous discretization alone. As such, the bending-strip method is viewed as a physically-motivated penalty technique. The bending-strip method was successfully employed in the modeling of wind turbine blades in [2,1].

Constraints are typically enforced by formulating them in a master-slave formalism that imposes the constraints sequentially when explicit time integration is used in structural dynamics. While feasible in many cases, a sequential imposition is not always possible, leading to the development of *consistent constraint explicit* methods [14] that append constraint equations to the mass matrix using Lagrange multipliers. Although this method is effective, it is also more expensive in comparison to standard explicit formulations. Constraints that cannot be analytically eliminated or imposed sequentially are therefore imposed using the penalty method. If a constraint approach is taken to enforce the continuity between patches, the bending strip is therefore preferred based on efficiency considerations.

#### 4 The blended shell element formulation

The blended shell formulation proposed here may be thought of as selectively adding rotational degrees of freedom instead of adding constraints. Although adding degrees of freedom to satisfy a set of constraints sounds counter intuitive, choosing generalized coordinates that automatically satisfy the required kinematic constraints has a long history in the analysis of mechanisms [29]. This approach eliminates the need to enforce the nonlinear  $C^1$  continuity constraints, eliminates the possibility of locally over constraining the structure at complicated intersections, and simplifies imposing the boundary conditions. More specifically, rotational degrees of freedom are selectively added to a rotation-free formulation [6] as needed to allow patches to be connected. This formulation is not restricted to the control points on the boundary, but permits the rotational degrees of freedom to be added anywhere within the patch to permit arbitrary connections. It also simplifies the imposition of the boundary conditions.

The blended formulation may be interpreted as one that switches from a Kirchhoff-Love theory to a Reissner-Mindlin theory at appropriate locations in the structure. With thin shell theories, the angle between two patches is rigidly fixed, but this

is not true for shear deformable theories. The ability to switch between theories therefore has some attractive properties for large deformation problems such as metal forming. One operation that is commonly performed on metal stamping is “flanging,” where the sheet is locally folded back on itself to stiffen the structure, introducing a  $C^0$  discontinuity along the fold line. This operation is commonly simulated in metal stamping analyses using standard  $C^0$  shell elements without any difficulty, but similar simulations would not be possible with a pure  $C^1$  formulation.

#### 4.1 The shell kinematics

The shell kinematics follow the degenerated solid approach to shells [17] where the volume is reduced from three dimensions to a two-dimensional representation. All coordinates are in the current configuration. By assumption, the thin dimension is associated with the third parametric coordinate  $s_3$ . For convenience,  $-1 \leq s_3 \leq +1$ . A reference surface  $\mathbf{x}^{\text{RS}}(s_1, s_2)$  midway between the upper and lower surfaces has the parameteric coordinate  $s_3 = 0$ . A fiber vector of unit length  $\hat{\mathbf{f}}(s_1, s_2)$  and a thickness function  $h(s_1, s_2)$  are introduced so that the coordinates of any point in the shell are defined in terms of the parametric coordinates as

$$\mathbf{x}(s_1, s_2, s_3) = \mathbf{x}^{\text{RS}}(s_1, s_2) + s_3 \frac{h}{2}(s_1, s_2) \hat{\mathbf{f}}(s_1, s_2) \quad (8)$$

In most shell formulations, the thickness function is assumed to be independent of time and therefore volume is not conserved for large membrane strains. Recent efforts to incorporate thickness strains by adding extra degrees of freedom have been very successful, e.g., [9,10]. Earlier efforts [16], where the normal strain required to satisfy the zero normal stress condition is integrated through the thickness to update the thickness, also work well. For notational simplicity, the thickness is assumed to be independent of time, and the extension of the current work to incorporate thickness changes may be introduced using any of the methods developed for standard shell elements.

The Kirchhoff-Love hypothesis defines  $\hat{\mathbf{f}}$  as the unit normal  $\mathbf{n}$  to the reference surface. The normal is defined from the derivatives of the midsurface,

$$\mathbf{p} = \frac{\partial \mathbf{x}}{\partial s_1} \times \frac{\partial \mathbf{x}}{\partial s_2}, \quad \text{and} \quad \mathbf{n} = \frac{\mathbf{p}}{|\mathbf{p}|}. \quad (9)$$

With this assumption, the motion of the shell volume is uniquely defined by the motion of the reference surface, and it, in turn, is defined entirely by the motion of the control points  $\mathbf{x}_A$ .

The Reissner-Mindlin hypothesis adds the coordinates defining the orientation of  $\hat{\mathbf{f}}$  as new solution variables, allowing the development of transverse shear strain. The

orientation of the fiber vector can be represented by the points on the surface of a sphere, giving two extra rotational degrees of freedom,  $\hat{\mathbf{f}}(\theta_1, \theta_2)$ . Working with two finite angles in practice, however, is difficult in structures with complicated shell intersections. Most shell element formulations use the simpler representation of the three degrees of freedom associated with the angular velocity  $\boldsymbol{\omega}$  about the global coordinate directions,

$$\dot{\hat{\mathbf{f}}} = \boldsymbol{\omega} \times \hat{\mathbf{f}}. \quad (10)$$

As a practical matter, many Reissner-Mindlin shell elements set the fiber vector to be the current configuration of the reference surface normal, e.g., [3], simplifying the formulation and increasing its robustness.

## 4.2 The discrete kinematics

The motion of the reference surface (denoted with the superscript RS) is commonly expressed as

$$\mathbf{x}^{\text{RS}} = \sum_A N_A \mathbf{x}_A, \quad \mathbf{u}^{\text{RS}} = \sum_A N_A \mathbf{u}_A, \quad (11)$$

where  $\mathbf{u}^{\text{RS}}$  is the displacement of the reference surface, but there are numerous choices for the discrete expression of the displacement contribution associated with the fiber vector. Most Kirchhoff-Love elements directly evaluate the normal using Equation 9 at  $(s_1, s_2)$ , i.e.,  $\hat{\mathbf{f}}(s_1, s_2) = \mathbf{n}(s_1, s_2)$  in Equation 8, giving the discrete expression for the coordinates,

$$\mathbf{x}(s_1, s_2, s_3) = \sum_A N_A(s_1, s_2) \mathbf{x}_A + s_3 \frac{h}{2}(s_1, s_2) \mathbf{n}(s_1, s_2). \quad (12)$$

The velocity field for the Kirchhoff-Love formulation is obtained by directly differentiating Equation 12,

$$\dot{\mathbf{x}} = \sum_A N_A \dot{\mathbf{x}}_A + s_3 \frac{h}{2} \dot{\mathbf{n}} \quad (13)$$

$$\dot{\mathbf{n}} = \frac{1}{|\mathbf{p}|} (\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) \dot{\mathbf{p}} \quad (14)$$

$$\mathbf{p} = \frac{\partial \mathbf{x}^{\text{RS}}}{\partial s_1} \times \frac{\partial \mathbf{x}^{\text{RS}}}{\partial s_2} \quad (15)$$

$$\dot{\mathbf{p}} = \frac{\partial \dot{\mathbf{x}}^{\text{RS}}}{\partial s_1} \times \frac{\partial \mathbf{x}^{\text{RS}}}{\partial s_2} + \frac{\partial \mathbf{x}^{\text{RS}}}{\partial s_1} \times \frac{\partial \dot{\mathbf{x}}^{\text{RS}}}{\partial s_2} \quad (16)$$

where the dependencies on the parametric coordinates are omitted for brevity.

Using the degenerated solid approach to formulate a shell, the fiber contribution is

interpolated from the nodes as in the Hughes-Liu element [17],

$$\hat{\mathbf{f}}(s_1, s_2) = \sum_A N_A(s_1, s_2) \hat{\mathbf{f}}_A, \quad (17)$$

and the resulting discrete representation of the coordinates is

$$\mathbf{x}(s_1, s_2, s_3) = \sum_A N_A(s_1, s_2) \left( \mathbf{x}_A + s_3 \frac{h}{2} \hat{\mathbf{f}}_A \right). \quad (18)$$

Our previous isogeometric shells [5,6] use Equation 18 with  $\hat{\mathbf{f}}_A = \mathbf{n}_A$  for robustness and simplicity.

Differentiating Equation 18 for the velocity field using the normal vector for the fiber direction gives

$$\dot{\mathbf{x}} = \sum_A N_A \left( \dot{\mathbf{x}}_A + s_3 \frac{h}{2} \dot{\mathbf{n}}_A \right). \quad (19)$$

For the Reissner-Mindlin isogeometric shells [5], the time derivative of the normal is evaluated as

$$\dot{\mathbf{n}}_A = \boldsymbol{\omega}_A \times \mathbf{n}_A, \quad (20)$$

the same expression used in standard  $C^0$  elements. The time derivative of the normal may also be evaluated by taking the time derivative of Equation 9 evaluated at  $A$  as in the rotation-free formulation [6],

$$\dot{\mathbf{n}}_A = \frac{1}{|\mathbf{p}_A|} (\mathbf{I} - \mathbf{n}_A \otimes \mathbf{n}_A) \dot{\mathbf{p}}_A \quad (21)$$

$$\mathbf{p}_A = \frac{\partial \mathbf{x}}{\partial s_1} \Big|_A \times \frac{\partial \mathbf{x}}{\partial s_2} \Big|_A \quad (22)$$

$$\dot{\mathbf{p}}_A = \frac{\partial \dot{\mathbf{x}}}{\partial s_1} \Big|_A \times \frac{\partial \mathbf{x}}{\partial s_2} \Big|_A + \frac{\partial \mathbf{x}}{\partial s_1} \Big|_A \times \frac{\partial \dot{\mathbf{x}}}{\partial s_2} \Big|_A \quad (23)$$

Note that “evaluated at  $A$ ” means that for every control point  $A$  there is a unique location in the parametric domain where  $\mathbf{n}_A$  is evaluated (see [5,6] for further details).

### 4.3 Selective enrichment with the blended formulation

The only difference between the kinematics of the rotation-free formulation [6] and the Reissner-Mindlin formulation [5] is the evaluation of  $\dot{\mathbf{n}}_A$ , as given by Equations 21 and 20, respectively. Since these equations refer only to control point variables, the equations may be invoked on a control point by control point basis. In effect, this shell formulation permits the smooth blending of a thin shell theory with a shear deformable one.

Rotational degrees of freedom may therefore be selectively introduced to permit the transmission of the bending moment along lines of  $C^0$  continuity without introducing constraints by locally using a shear deformable shell theory. This is accomplished by using the kinematics defined by Equations 18 and 19, and choosing  $\dot{\mathbf{n}}$  to be defined by Equation 21 for the control points in the thin shell regions,  $A \in \mathcal{T}$ , and by Equation 20 for control points in the shear deformable regions,  $A \in \mathcal{S}$ ,

$$\mathbf{x} = \sum_A N_A \left( \mathbf{x}_A + s_3 \frac{h}{2} \mathbf{n}_A \right) \quad (24)$$

$$\begin{aligned} \dot{\mathbf{x}} = & \sum_A N_A \dot{\mathbf{x}}_A \\ & + s_3 \frac{h}{2} \left( \sum_{A \in \mathcal{T}} N_A \frac{1}{|\mathbf{p}_A|} (\mathbf{I} - \mathbf{n}_A \otimes \mathbf{n}_A) \dot{\mathbf{p}}_A + \sum_{A \in \mathcal{S}} N_A \boldsymbol{\omega}_A \times \mathbf{n}_A \right) \end{aligned} \quad (25)$$

Elements with control points belonging to both sets exhibit a behavior somewhere between the pure thin shell and shear deformable shell theories. Example problems presented later explore the influence of using a shear deformable theory in relatively localized regions of an otherwise thin shell analysis.

#### 4.4 Implementation

The implementation is a combination of the Reissner-Mindlin [5] and rotation-free [6] isogeometric shell implementations. These shells, in turn, follow the generalized element strategy [4]. The focus here is on how much of the various required calculations can be carried out in common in order to both maximize the efficiency of the implementation and define where independent operations must be performed.

While the primary motivation for introducing rotations is joining patches in a multi-patch analysis, they are also convenient for imposing boundary conditions. Additionally, there are occasions when having  $C^0$  lines running through a patch simplifies the model generation. To incorporate these extensions, the definition of the set  $\mathcal{S}$  is generalized to include any control point with rotational degrees of freedom even if they are not on the patch boundary, and the set  $\mathcal{T}$  contains the remainder. Every element may therefore contain an arbitrary combination of control points with, and without, rotational degrees of freedom.

Gauss quadrature is performed over the reference surface spanned by  $s_1$  and  $s_2$ , and through the thickness direction spanned by  $s_3$ . Many of the common terms are not functions of  $s_3$ . Efficiency is therefore enhanced by evaluating them once before performing the integration loop through the thickness, and making the integration over the reference surface the outer loop. Variables associated with the outer and inner integration loops are given the superscripts  $G$  and  $g$ , respectively.

The integration point within an element is, for example, specified by the parametric coordinates  $(s_1^G, s_2^G, s_3^g)$ , and the volume integration is evaluated as

$$V = \int_{\mathbf{x}} d\mathbf{x} = \int_{\mathbf{s}} \det(\mathbf{J}) d\mathbf{s} = \sum_G \sum_g W^G w^g \det(\mathbf{J}^{Gg}) = \sum_G \sum_g V^{Gg}, \quad (26)$$

where the integration weights for  $G$  and  $g$  are  $W^G$  and  $w^g$ , respectively. For compactness, the evaluation of a function at the specific integration point  $(s_1^G, s_2^G, s_3^g)$  is denoted by the dual superscripts “ $Gg$ ” and the volume contribution of integration point  $Gg$  is denoted  $V^{Gg}$ .

#### 4.5 Evaluation of the normals and their time derivatives

Since the basis functions are not interpolatory, there is no natural location for evaluating the normal. The derivatives of  $\mathbf{x}$  and  $\dot{\mathbf{x}}$  with respect to the two parametric coordinates  $s_1$  and  $s_2$  in Equations 15 and 16 are replaced by the vectors  $\mathbf{t}_i$  and  $\dot{\mathbf{t}}_i$

$$\mathbf{t}_{Ai} = \sum_B C_{Bi}^A \mathbf{x}_B \quad (27)$$

$$\dot{\mathbf{t}}_{Ai} = \sum_B C_{Bi}^A \dot{\mathbf{x}}_B \quad (28)$$

where  $C_{Bi}^A$  are evaluated by a lifting operation (for details and explicit formulae, see [6]). The normals are computed at each control point outside of the element integration loop since they are independent of the integration points,

$$\mathbf{p}_A = \mathbf{t}_{A1} \times \mathbf{t}_{A2}, \quad \mathbf{n}_A = \frac{\mathbf{p}_A}{|\mathbf{p}_A|}, \quad \text{no sum on } A. \quad (29)$$

The time derivative of the normals are dependent on the degrees of freedom at the control point,

$$\dot{\mathbf{n}}_A = \boldsymbol{\omega}_A \times \mathbf{n}_A, \quad A \in \mathcal{S} \quad (30)$$

$$\dot{\mathbf{n}}_A = \frac{1}{|\mathbf{p}_A|} (\mathbf{I} - \mathbf{n}_A \times \mathbf{n}_A) \dot{\mathbf{p}}_A, \quad A \in \mathcal{T} \quad (31)$$

$$\dot{\mathbf{p}}_A = \dot{\mathbf{t}}_{A1} \times \mathbf{t}_{A2} + \mathbf{t}_{A1} \times \dot{\mathbf{t}}_{A2} \quad (32)$$

Evaluating the time derivative of the normal once outside the element integration loops simplifies the evaluation of the velocity gradient, which is evaluated at every integration point.

#### 4.6 Evaluation of the Jacobian and the velocity gradient

The summation convention is used as follows: When repeated, it is to be understood that subscripts  $i$ ,  $j$ ,  $k$ , and  $\ell$  are summed over 1, 2, and 3, unless it is explicitly indicated to sum over 1 and 2.

The Jacobian  $\mathbf{J}$  and velocity gradient  $\mathbf{L}$  are functions of the control point coordinates, the normal, and the time derivative of the normal, and they are therefore evaluated without regard to the degrees of freedom at the control points. Note that both are affine functions of  $s_3$ , allowing most of the calculations to be performed once at each reference surface integration point.

$$J_{ik}^G(s_3) = \frac{\partial x_i}{\partial s_k} = J_{ik}^{G1} + s_3 J_{ik}^{G2} \quad (33)$$

$$J_{ik}^{G1} = \sum_A \frac{\partial N_A}{\partial s_k} x_{Ai} \quad k = 1, 2 \quad (34)$$

$$J_{i3}^{G1} = \sum_A N_A \frac{h_A}{2} n_{Ai} \quad (35)$$

$$J_{ik}^{G2} = \sum_A \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} n_{Ai} \quad k = 1, 2 \quad (36)$$

$$L_{ij}^G(s_3^g) = L_{ij}^{G1} + s_3 L_{ij}^{G2} \quad (37)$$

$$L_{ij}^{G1} = \sum_{k=1,2} J_{kj}^{-1} \left[ \sum_A \frac{\partial N_A}{\partial s_k} \dot{x}_{Ai} \right] + J_{j3}^{-1} \left[ \sum_A N_A \frac{h_A}{2} \dot{n}_{Ai} \right] \quad (38)$$

$$L_{ij}^{G2} = \sum_{k=1,2} J_{kj}^{-1} \left[ \sum_A \frac{\partial N_A}{\partial s_k} \frac{h_A}{2} \dot{n}_{Ai} \right]. \quad (39)$$

#### 4.7 Stress evaluation

The stress is updated in a co-rotational formulation similar to the one used by Beltrachko and Tsay [3]. It is completely independent of the blending, and briefly documented here only for completeness. A local co-rotational coordinate system  $\mathbf{e}_i^\ell, i = 1, 2, 3$  is defined on the reference surface and is used for all the integration points through the thickness. The normal direction defines  $\mathbf{e}_3^\ell$ , and the other two directions are calculated using the procedure of Hughes and Liu [17,15]. The rotation matrix from the local to the global coordinate system,  $\mathbf{v}^g = \mathbf{R}\mathbf{v}^\ell$ , is

$$\mathbf{R} = \begin{bmatrix} \mathbf{e}_1^\ell & \mathbf{e}_2^\ell & \mathbf{e}_3^\ell \end{bmatrix}. \quad (40)$$



The rate of deformation  $\mathbf{D}$  in the global system is calculated from the velocity gradient, and the local rate  $\mathbf{D}^\ell$  is obtained by

$$\mathbf{D}^{Gg} = \frac{1}{2} \left( \mathbf{L}^{Gg} + (\mathbf{L}^{Gg})^T \right), \quad \mathbf{D}^\ell = \mathbf{R}^T \mathbf{D}^{Gg} \mathbf{R}. \quad (41)$$

The normal component  $D_{33}^\ell$  calculated in this manner will, in general, not result in the normal stress condition  $\sigma_{33}^\ell = 0$  being satisfied. A simultaneous solution for  $\sigma_{33}^\ell(D_{33}^\ell) = 0$  and the stress update is performed using the appropriate algorithm [28,14,26]. Finally, the updated stress is rotated into the global coordinate system at the end of the time step,

$$\boldsymbol{\sigma}^{Gg} = \mathbf{R} \boldsymbol{\sigma}^\ell \mathbf{R}^T. \quad (42)$$

#### 4.8 Evaluation of the residual

To preserve the commonality between the two different types of control points for as long as possible, the virtual velocity is expressed in terms of the normal and differentiated to obtain the expression for the virtual gradient,

$$\frac{\partial \delta \dot{\mathbf{x}}}{\partial \mathbf{x}} = \sum_A \left\{ \frac{\partial N_A}{\partial \mathbf{x}} \delta \dot{\mathbf{x}}_A + \frac{h_A}{2} \left( s_3 \frac{\partial N_A}{\partial \mathbf{x}} + N_A \frac{\partial s_3}{\partial \mathbf{x}} \right) \delta \dot{\mathbf{n}}_A \right\}. \quad (43)$$

Index notation simplifies gathering the common terms in the sequel,

$$\begin{aligned} \frac{\partial \delta \dot{x}_i}{\partial x_j} &= \sum_A \left\{ \frac{\partial N_A}{\partial x_j} \delta \dot{x}_{Ai} + \frac{h_A}{2} \left( s_3 \frac{\partial N_A}{\partial x_j} + N_A \frac{\partial s_3}{\partial x_j} \right) \delta \dot{n}_{Ai} \right\} \\ &= \sum_A \left\{ \sum_{k=1,2} (J^{Gg})_{kj}^{-1} \frac{\partial N_A}{\partial s_k} \delta \dot{x}_{Ai} + \frac{h_A}{2} (J^{Gg})_{kj}^{-1} \left( s_3 \frac{\partial N_A}{\partial s_k} + N_A \delta_{3k} \right) \delta \dot{n}_{Ai} \right\} \end{aligned} \quad (44)$$

where  $\mathbf{J}^{-1}$  is the inverse of the Jacobian matrix.

Numerically integrating separately in the reference plane  $s_1 - s_2$  and through the thickness direction  $s_3$ , the virtual work associated with the stress is

$$\begin{aligned} \delta W &= \sum_A \sum_{Gg} \sum_{k=1,2} \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} \frac{\partial N_A^G}{\partial s_k} \delta \dot{x}_{Ai} V^{Gg} \\ &\quad + \sum_A \sum_{Gg} \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} \frac{h_A}{2} \left( s_3^g \frac{\partial N_A^G}{\partial s_k} + N_A^G \delta_{3k} \right) \delta \dot{n}_{Ai} V^{Gg} \end{aligned} \quad (45)$$

Separating the terms based on their dependencies on  $G$  and  $g$ , the virtual work is

$$\begin{aligned}
\delta W = & \sum_A \delta \dot{x}_{Ai} \sum_G \sum_{k=1,2} \frac{\partial N_A^G}{\partial s_k} \sum_g \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} V^{Gg} \\
& + \sum_A \frac{h_A}{2} \delta \dot{n}_{Ai} \left\{ \sum_G \sum_{k=1,2} \frac{\partial N_A^G}{\partial s_k} \sum_g s_3^g \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} V^{Gg} \right. \\
& \left. + \sum_G N_A^G \sum_g \sigma_{ij}^{Gg} (J^{Gg})_{3j}^{-1} V^{Gg} \right\}
\end{aligned} \tag{46}$$

Collecting the common terms that are summed over  $g$  defines two resultant vectors,

$$R_{ik}^G = \sum_g \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} V^{Gg} \quad \text{and} \quad \hat{R}_{ik}^G = \sum_g s_3^g \sigma_{ij}^{Gg} (J^{Gg})_{kj}^{-1} V^{Gg}. \tag{47}$$

Substituting them into Equation 46 gives the forces that are work conjugate to  $\delta \dot{x}$  and  $\delta \dot{n}$ ,

$$\bar{F}_{Ai}^x = \sum_G \sum_{k=1,2} R_{ik}^G \frac{\partial N_A^G}{\partial s_k} \tag{48}$$

$$\bar{F}_{Ai}^n = \frac{h_A}{2} \sum_G \left( \sum_{k=1,2} \hat{R}_{ik}^G \frac{\partial N_A^G}{\partial s_k} + R_{i3}^G N_A^G \right) \tag{49}$$

Note that the residual evaluation, consisting of the summation over the two nested loops for the integration, is independent of whether a control point has rotational degrees of freedom or not to this point. The only operations that depend on the degrees of freedom are performed outside of the integration loop just before the element residual assembly.

For the rotation-free control points, the discrete virtual power equation at control point  $A$  gives

$$F_{Ai}^x = \bar{F}_{Ai}^x + \sum_B \bar{F}_{Bj}^n \frac{\partial \dot{n}_{Bj}}{\partial \dot{x}_{Ai}}. \tag{50}$$

Introducing the projection matrix  $P_{Aij}$  and the common terms  $C_{Bk}^A$  (see Equation 27), the final expression for the control point translational forces is

$$F_{Ai}^x = \bar{F}_{Ai}^x + \sum_B e_{ik\ell} \bar{F}_{Bj}^n P_{Bjk} C_{A\ell}^B \tag{51}$$

where

$$P_{Aij} = \frac{1}{|\mathbf{p}_A|} (\delta_{ij} - n_{Ai} n_{Aj}) \tag{52}$$

For the control points with rotational degrees of freedom, the discrete virtual power

equations at control point  $A$  are

$$F_{Ai}^x = \bar{F}_{Ai}^x \quad \text{and} \quad F_{Ai}^\omega = \bar{F}_{Aj}^n \frac{\partial \dot{n}_{Aj}}{\partial \omega_{Ai}} = e_{ikj} n_{Ak} \bar{F}_{Aj}^n. \quad (53)$$

## 5 Example calculations

Explicit finite element formulations traditionally rely on lower-order elements with uniformly reduced integration for their computational efficiency. Therefore, explicit analysis typically uses elements with linear basis functions, one point integration, and hourglass control. Higher order Lagrange elements are not practical for explicit analysis because of the increased cost of evaluating the element, their instability under large deformations, and their reduced time step size due the large errors in their highest frequencies. The convergence of NURBS basis functions at their highest frequencies eliminates the time step size penalty, but the cost of evaluating the elements still increases with  $p$ . Based on previous benchmarks [4], quadratic NURBS elements with uniform reduced integration offer the greatest opportunity for enhancing the spatial accuracy without increasing the analysis cost in comparison to current formulations. Throughout, each element is integrated with the  $2 \times 2$  uniformly reduced Gauss quadrature rule in the plane, and the three-point rule through the thickness. Indeed, as shown previously [4] and in the last example in this section, isogeometric analysis with quadratic NURBS offers the possibility of increasing accuracy at a reduced computational cost. The focus in the examples presented here is therefore on quadratic NURBS exclusively. However, we note that recent work on isogeometric collocation methods in explicit dynamics suggests that this picture may change.

### 5.1 Nonlinear L-beam

In order to demonstrate the ability of the current formulation to preserve the initial angle of multiple patches along  $C^0$  lines, an L-shaped cantilever beam subjected to a point load is chosen. This example has been analyzed by Kiendl et al. [20,21] to show the performance of the constraint equations and the bending strip method in multi-patch analysis.

Quadratic NURBS elements in three different model configurations are analyzed as shown in Figure 2. Starting from the left, the first model has two patches with rotation-free isogeometric shell elements everywhere. The second model is split into two patches with the blended isogeometric shell formulation with the rotational degrees of freedom added along the  $C^0$  edge. Finally, a two patch model with the Reissner-Mindlin isogeometric shell formulation [5] is also analyzed. The structural response is shown in Figure 3. The first model, as expected, illustrates the

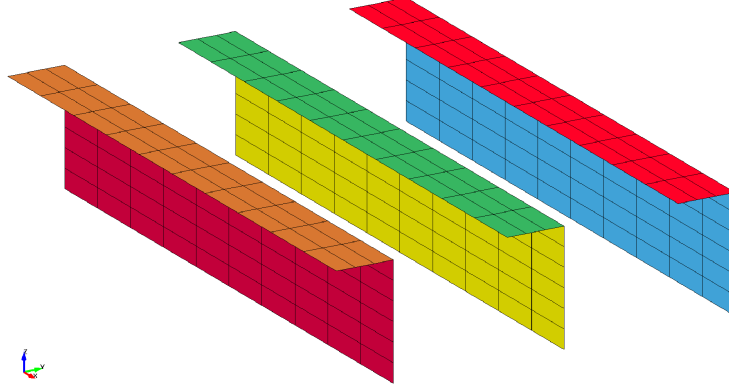


Fig. 2. Initial meshes of the L-beams modeled with quadratic NURBS. From left to right: two patches of rotation-free elements, two patches of blended elements with rotations along the fold line, and finally two patches of Reissner-Mindlin shell elements.

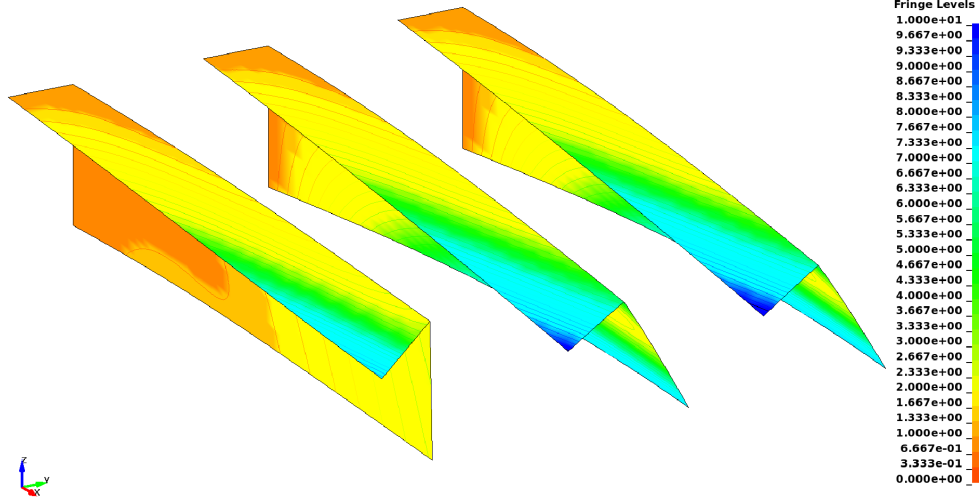


Fig. 3. Final configuration of the L-beams. From left to right: two patches of rotation-free elements, two patches of blended elements with rotations along the fold line, and finally two patches of Reissner-Mindlin shell elements.

inability of rotation-free shells to transmit bending moments across  $C^0$  lines. The  $C^0$  line behaves like a hinge and the initial  $90^\circ$  angle is not maintained. The other models give identical results, demonstrating the proper transmission of bending moments with the blended formulation.

## 5.2 Vibration of a thin simply-supported square plate

The previous example evaluated the accuracy of different methods for multi-patch analysis subject to single modes of deformation. An eigenvalue analysis introduces a large number of modes of deformation with the complexity of the deformations increasing with the frequency, giving additional insight. This example was used to

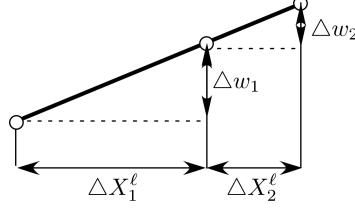


Fig. 4. The geometry of the local constraint equation. The  $C^0$  interface is normal to the plane of the figure and located at the middle control point,  $\Delta w$  is in the global coordinate system perpendicular to the plate, and  $X^\ell$  is in the plane of the plate and perpendicular to the  $C^0$  boundary.

test isogeometric analysis in [13], and was also analyzed in [6] to demonstrate the accuracy of the rotation-free isogeometric shell formulation in linear analysis. The exact eigenvalues in radians for a square plate of length  $L$  and thickness  $h$ , using thin plate theory, are

$$\omega_{ij} = C \left( i^2 + j^2 \right) \quad 0 < i, j \quad (54)$$

$$C = \pi^2 \sqrt{\frac{E}{\rho (12 (1 - \nu^2))}} \frac{h}{L^2} \quad (55)$$

where  $E$  is Young's modulus,  $\nu$  is Poissons's ratio, and  $\rho$  is the density. In this example, the values are chosen to be  $E = 10^7$ ,  $\nu = 0.3$ ,  $\rho = 1.0$ ,  $h = 0.05$ ,  $L = 10$ .

The square plate is evenly divided into two patches and the eigenvalue problem is solved using three formulations:

- (1) The rotation-free-formulation [6] with the geometric continuity enforced with linear constraints formulated in a manner similar to Kiendl et al. [21]. The constraints are discussed in more detail below. This formulation is referred to as RF1.
- (2) A second rotation-free formulation, based on the updated Lagrangian formulation given by Equation 12 with  $C^1$  continuity enforced using the same linear constraints as in (1). The two rotation-free formulations differ in where their normals are evaluated, however they give the same eigenvalues on a single patch for the square plate problem. Of interest here is the sensitivity of the eigenvalue accuracy to the different rotation-free formulations when using the constraints to enforce continuity. This formulation is referred to as RF2.
- (3) The blended formulation with the common boundaries between the patches having translational and rotational degrees of freedom while the interior of the patches is rotation-free. Since constraints on the rotations are not imposed on the perimeter of the plate, the perimeter control points have only the three constrained translational degrees of freedom.

The linear constraints enforcing the continuity [21] are defined in terms of the local reference geometry and the infinitesimal displacements as illustrated in Figure 4. Neglecting the higher order terms, the linear constraint equation in the local co-

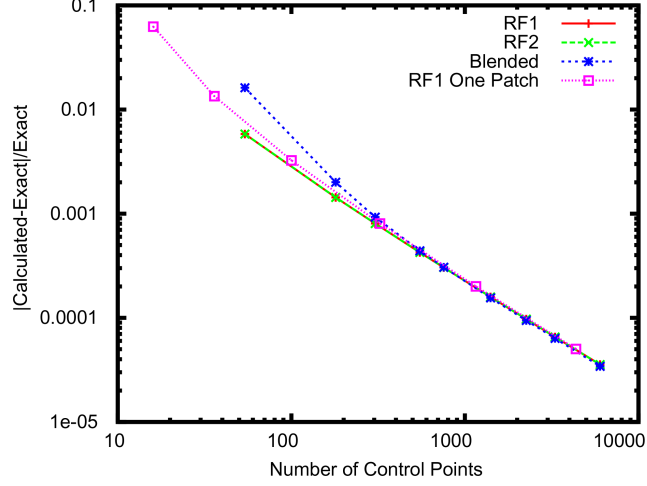


Fig. 5. Convergence of the first eigenvalue for the two patch mesh of the square plate problem as a function of the number of control points.

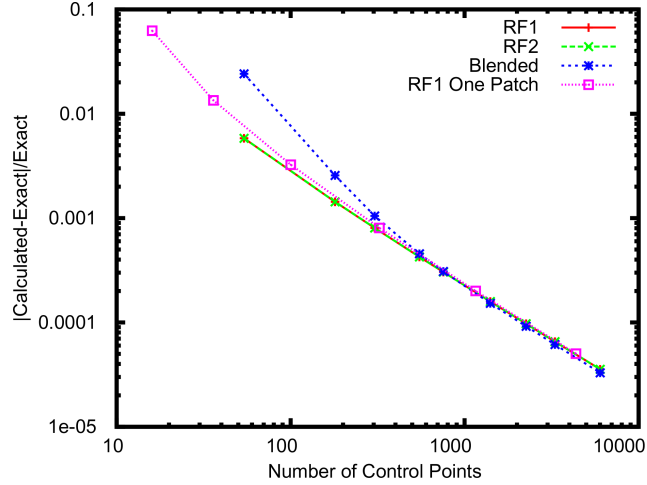


Fig. 6. Convergence of the first eigenvalue for the four patch mesh of the square plate problem as a function of the number of control points.

ordinate system defined at the shared boundary of two coplanar patches simplifies to

$$\Delta X_1^\ell \Delta w_2 - \Delta X_2^\ell \Delta w_1 = 0. \quad (56)$$

where  $\mathbf{X}^\ell$  is in the local coordinate system of the flat plate with one axis aligned with the patch boundary.

The convergence rates for the first eigenvalue, using two quadratic NURBS patches, are shown in Figure 5.

An additional  $C^0$  discontinuity is introduced by splitting the domain into four square patches. This problem is fundamentally different from the previous one because it has intersecting  $C^0$  lines involving nine control points in the constraints at the center of the plate. If the constraint patterns along the two  $C^0$  lines are blindly imposed along the  $C^0$  lines, the nine control points are subjected to six constraint

equations, producing a singular system. Deleting both constraints for the center control point improved the accuracy of some modes and made the accuracy of others worse compared to the single patch results, but deleting only one restored the accuracy to almost exactly the single patch results. Constraints making the center control point displacement the average of a set of the surrounding control points,

$$w_C - \frac{1}{N} \sum_{A \in \mathcal{C}} w_A = 0, \quad (57)$$

where  $C$  designates the central control point and  $\mathcal{C}$  is a subset of the control points surrounding  $C$ , worked equally well. General, accurate, robust strategies for constraining control points at arbitrary shell intersections are currently unavailable.

No special treatment was required for the blended formulation. The center node was treated like any other node at a  $C^0$  line, i.e., it had a full set of rotational degrees of freedom. The control points on the simply supported boundary again had only the three constrained translational degrees of freedom.

The convergence rates are shown in Figure 6. These results are virtually identical to the results for one and two patches.

### 5.3 Gravity-loaded L-shaped plate

An L-shaped plate is loaded by gravity. The problem is specified numerically and the geometry, material, and boundary condition data are provided in Figure 8. The parameters are chosen to make this a nonlinear, finite deformation problem with coupled membrane and bending behavior. The plate is discretized with NURBS, and two meshes, coarse and fine, are shown in Figure 9. The parameterization has a discontinuous derivative along the line of symmetry. As a result, the underlying discretization is only  $C^0$ -continuous along the line of symmetry and a direct application of a rotation-free shell formulation is not possible in this case.

One approach to overcome this difficulty is to use the bending strip method. A bending strip patch may be generated along the symmetry line. A typical element of that patch is depicted in Figure 10. The figure reveals the difficulty with the application of the bending strip method in this case: Due to the abrupt change in the parameterization near the symmetry line, the covariant basis vector  $\mathbf{G}_1$ , which is typically used to define the direction of the bending stiffness operator, is rapidly changing direction from one side of the bending strip to the other. Furthermore, almost everywhere  $\mathbf{G}_1$  has a component parallel to the interface, which will incorrectly stiffen the structure in this direction. Despite this difficulty, the bending strip method was made to work in this case by integrating the contribution of the bending strip terms with one-point quadrature placed in the middle of the element. At this quadrature point the covariant basis vectors are orthogonal, and  $\mathbf{G}_1$  points in the di-

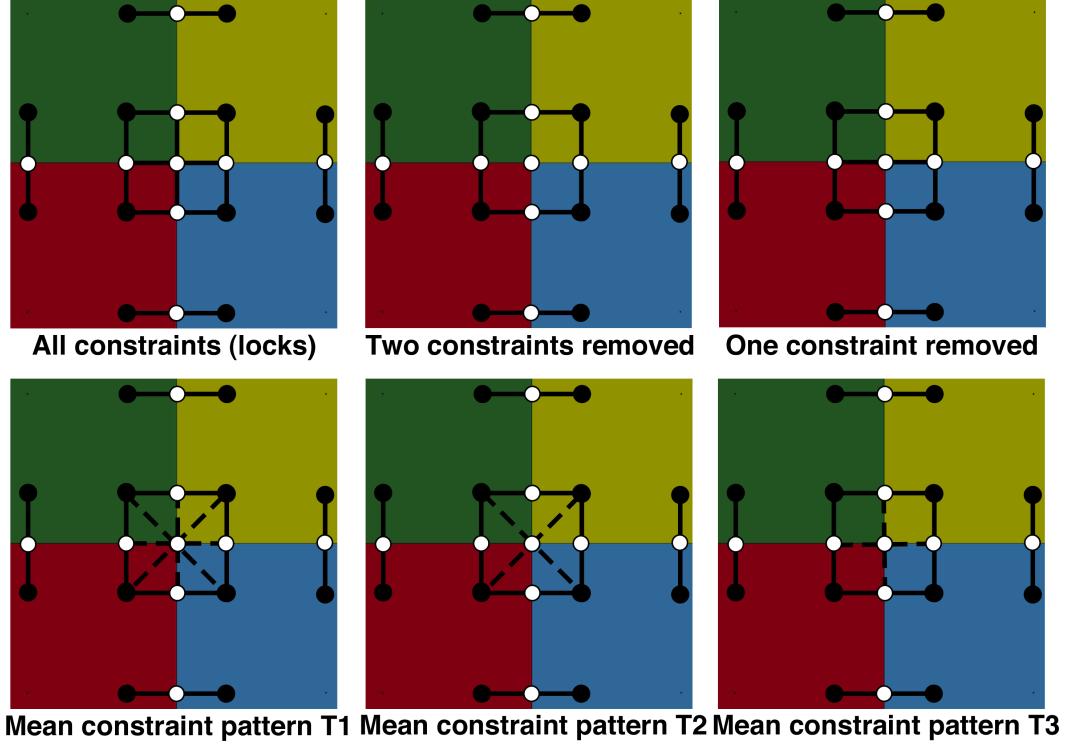


Fig. 7. Constraint configurations for the four-patch mesh. Solid lines indicate constraints of the type defined by Equation 56 and dashed lines, Equation 57. Control points shared between adjacent patches are indicated with white dots, and those associated with only one patch, black dots. The configuration with all constraints locks. With only one constraint removed, the model works correctly. The three patterns in the second row using Equation 57 worked equally well.

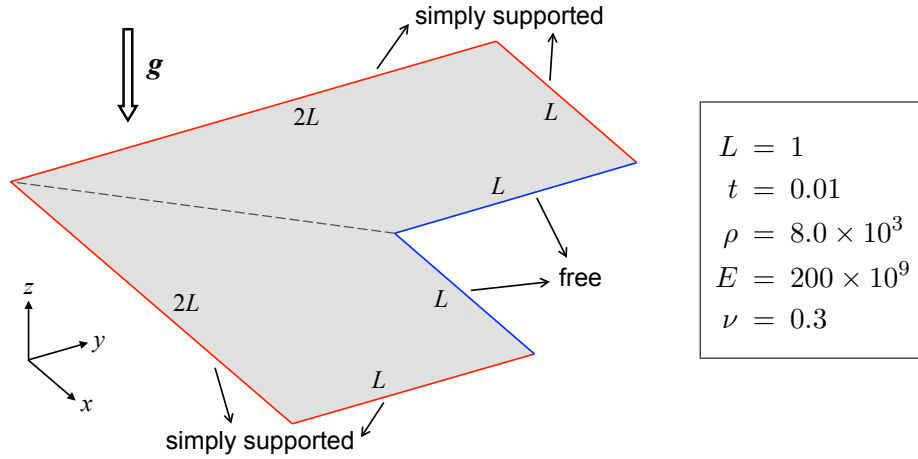


Fig. 8. L-shaped plate. Problem setup and data.

rection orthogonal to the interface. Figure 11 shows the results of the computations with and without the bending strip.

This “fix” is somewhat *ad-hoc*, and, in what follows, the L-shaped plate example is used to illustrate the good behavior of the proposed blended formulation, which



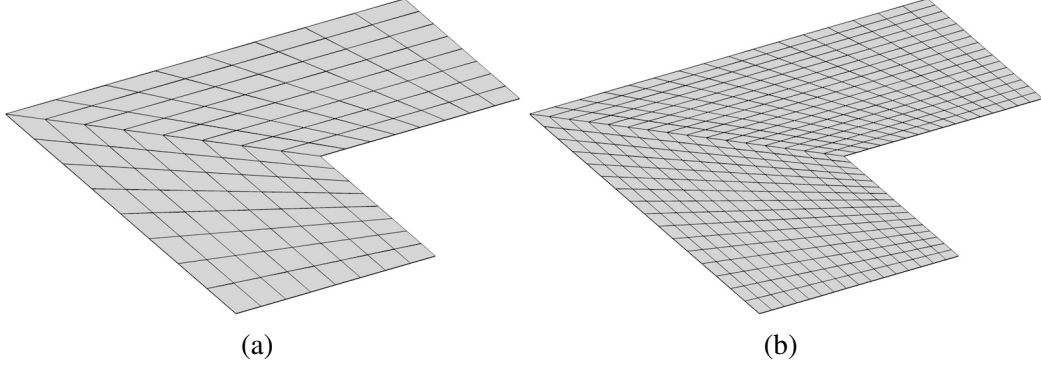


Fig. 9. L-shaped plate. NURBS meshes. Note that the parameterization of the domain has a discontinuous derivative along the symmetry line.

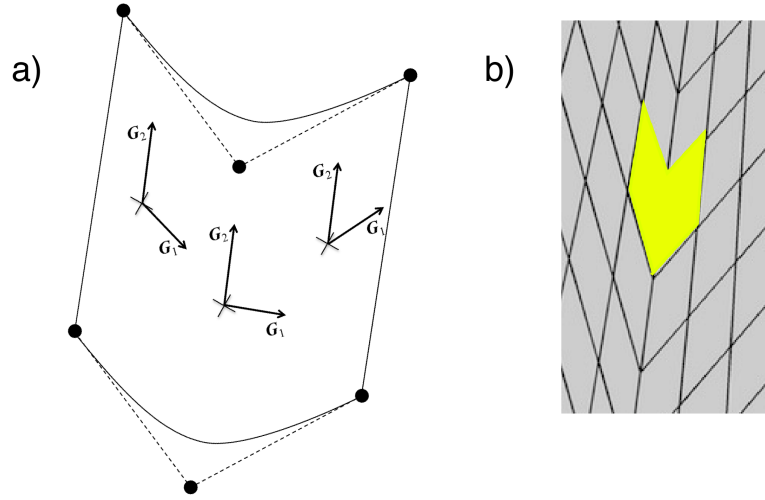


Fig. 10. L-shaped plate. Illustration of the difficulty with the application of the bending strip method to this example. The direction of the covariant basis vector  $\mathbf{G}_1$ , which is typically taken as the direction for the bending stiffness operator, is only truly orthogonal to the interface at the midpoint of the parametric line. Also note that the covariant basis vectors are orthogonal only at this location. The six control point element in a) overlays the two yellow elements in b).

does not rely on a good parameterization near  $C^0$ -continuous interfaces.

An analytical solution is not available for this problem. In a one-patch convergence study with  $p = 4$ , the maximum displacement converged to 0.003277, the value we used in lieu of an exact solution in Figure 12. The rotation-free solutions (the bending strip method uses a rotation-free shell [20]) converge from the soft side, while those with rotational degrees of freedom converge from the stiff side.

To evaluate the benefits of the blended formulation for implicit problems, the problem was also solved using the Reissner-Mindlin formulation [5] on the finest mesh,

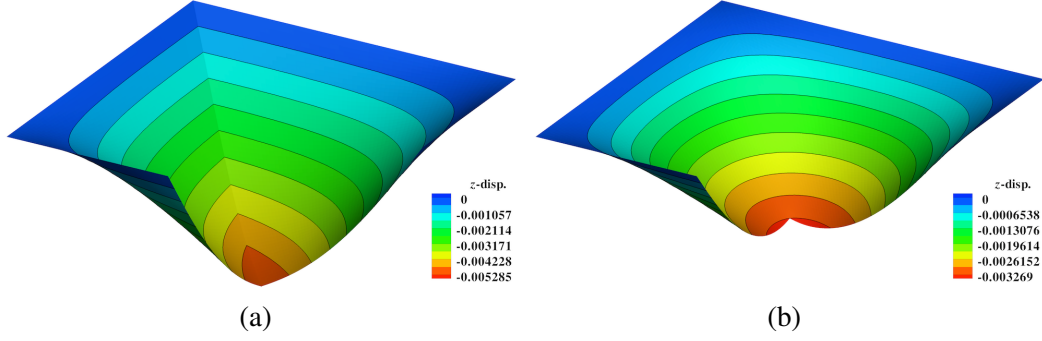


Fig. 11. L-shaped plate. Deformed configuration (scale factor of 200 is employed) colored by the  $z$ -displacement. (a) Result without the bending strip. (b) Result with the bending strip integrated using one-point quadrature. The computation without the bending strip gives an unphysical “kink” at the interface, while the one-point-quadrature bending strip formulation gives a physically correct answer.

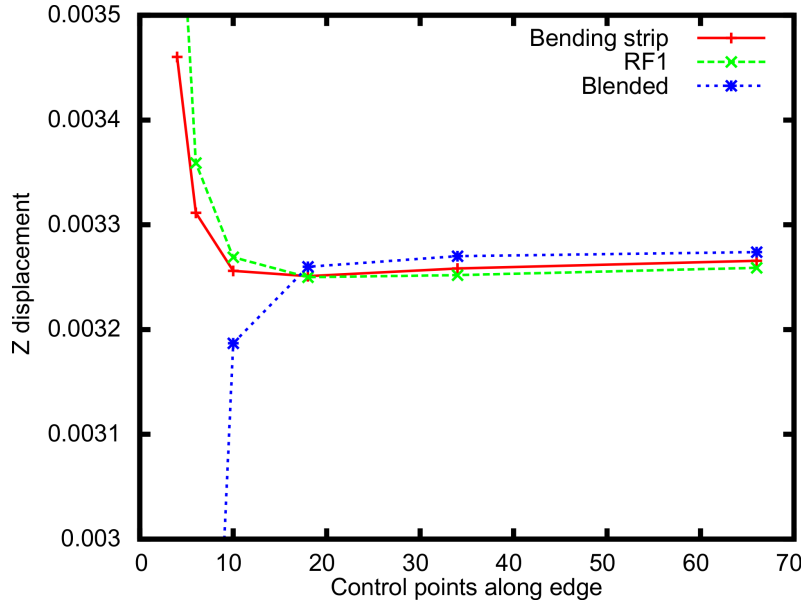


Fig. 12. Convergence of the  $z$  displacement at the origin for the L-shaped domain.

and the results are summarized in Table 1. The reported CPU times were obtained with the standard Linux timing function, and are accurate to approximately 0.01 second. To solve the nonlinear problem required one stiffness factorization and six solves. The storage is as expected: The blended formulation has half the number of degrees of freedom and requires one quarter the storage. For a flat plate, the Reissner-Mindlin formulation is effectively a five degree of freedom formulation, with the normal rotations assigned a small values in the stiffness matrix to avoid being singular. Although the concept of a “bandwidth” is not appropriate for a sparse solver, the logically regular structure of the mesh leads to an ordering such that the ratio of the Reissner-Mindlin to the blended formulation factorization costs is very close to  $(5/3)^3 = 4.63$ , and for the solves, the ratio is roughly  $(5/3)^2 = 2.78$ . Although the Reissner-Mindlin formulation has simpler terms for the stiffness ma-

Table 1

Comparison of Implicit Reissner-Mindlin and Blended Formulation Solution Costs

	Reissner-Mindlin	Blended	Ratio (R-M)/B
Number of equations	51093	25353	2.01
Memory ( $10^3$ 8-byte words)	3735	937	3.98
Element CPU (sec.)	7.37	5.65	1.30
Assembly CPU (sec.)	1.29	1.09	1.18
Factorization CPU (sec.)	12.7	2.74	4.63
Total Solve CPU (sec.)	0.27	0.09	3.00
Total CPU(sec.)	21.63	9.57	2.26

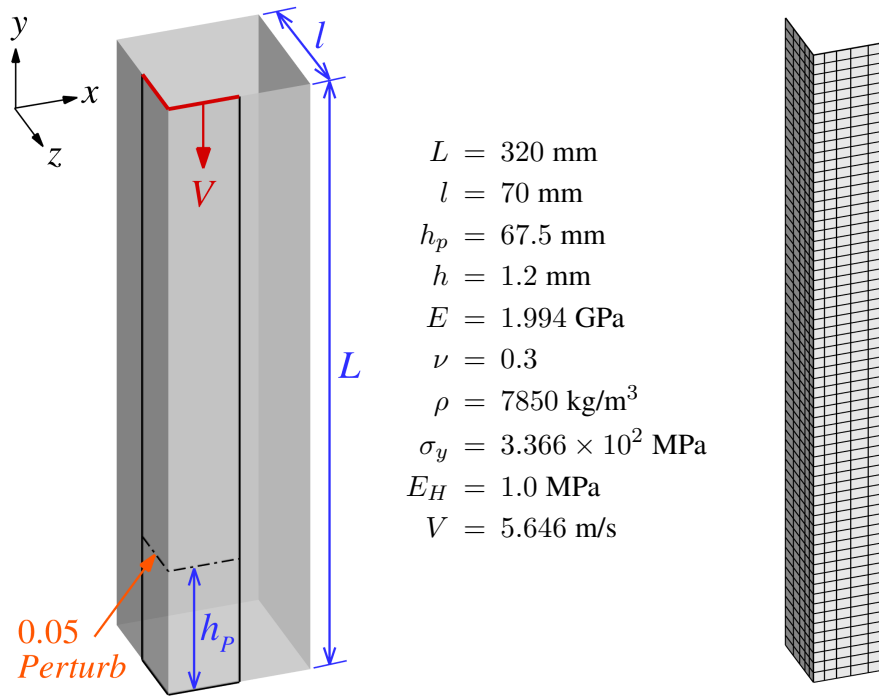


Fig. 13. Buckling of a square tube: problem description and the mesh. Only one quarter of the geometry is modeled with appropriate symmetry boundary conditions. Reprinted from [5]

trix and the residual, having more of them puts it at a disadvantage in terms of the overall element cost. The total gain in speed is over a factor of two for this modest sized problem.

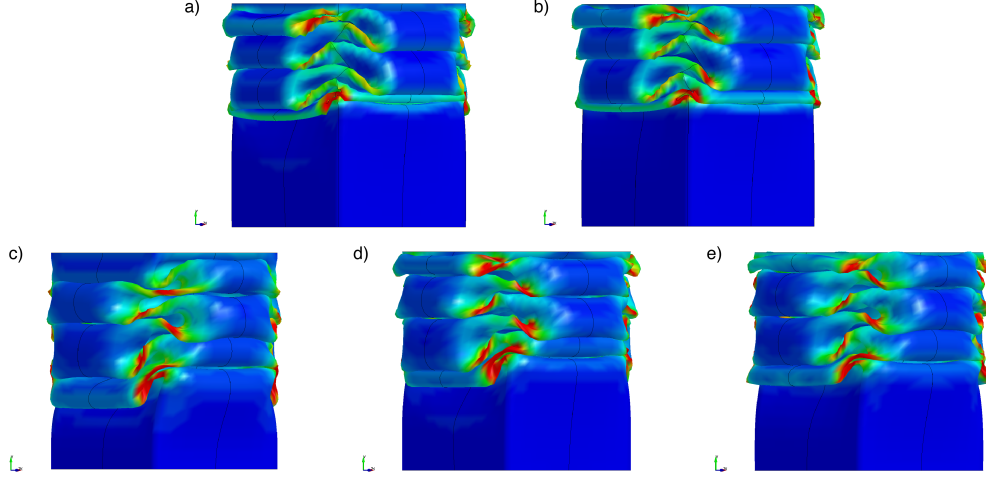


Fig. 14. Final configuration of the square tube buckling problem using a) the Belytschko-Tsay element, b) the fully-integrated quadrilateral shell element, c) the  $p = 2$  isogeometric Reissner-Mindlin shell, d) the  $p = 2$  rotation-free isogeometric shell with the continuity across the  $C^0$  lines enforced with rigid bodies, and e) the blended element.

Table 2

Square tube buckling benchmark summary

Element Type	Number of Elements	Number of Time Steps	CPU Seconds
1-Point Quad	2560	352631	833
Full Int. Quad	2560	414148	2861
Isogeo. RM	640	153596	582
Isogeo. RF	640	173706	835
Isogeo. blended	640	158192	754

#### 5.4 Buckling of a square tube

The problem of accordion-mode buckling of a square tube [8] was analyzed previously [5] to evaluate quadratic and quartic NURBS-based isogeometric Reissner-Mindlin shell elements. The problem definition, geometry, material parameters, and the NURBS mesh for one quarter of the domain are shown in Figure 13. An isotropic elastic-plastic material with linear plastic hardening is used to model the material response. The deformation is driven with a constant velocity at one end of the tube with the other end fixed. A geometric imperfection with an amplitude of 0.05 mm triggers the buckling at a height of 67.5 mm from the base.

One quarter of the tube is modeled using two quadratic NURBS patches that share the control points along the common edge, leading to 640 elements. Appropriate symmetry boundary conditions are used together with a standard single surface

contact algorithm in LS-DYNA [14,8] applied to the automatically generated interpolation elements [4]. Each quadratic NURBS element is subdivided into  $2 \times 2$  interpolation elements.

Three different NURBS-based isogeometric shell formulations were analyzed: The Reissner-Mindlin shell [5], the rotation-free shell [6], and the blended shell. Continuity for the rotation-free formulation is enforced between the patches using rigid bodies [7] composed of the common control point between adjacent patches and the first interior control points for each patch, and in a similar manner, their rotational boundary conditions at the upper and lower boundaries were imposed with rigid bodies composed of the boundary nodes and the first rows of nodes interior to the mesh. For the blended formulation, the rotational degrees of freedom are placed along all boundaries of the two NURBS patches.

The final deformations for the three shell formulations, and the LS-DYNA solutions using the Belytschko-Tsay element [3] and the type 16 fully-integrated, 4-node shell, are shown in Figure 14. The meshes for the standard elements have 2560 elements, four times as many as the quadratic NURBS. Using reduced  $2 \times 2$  integration with the quadratic NURBS elements results in the three isogeometric models having the same number of integration points as the Belytschko-Tsay model, while the fully-integrated shell has four times as many. The final configurations are remarkably similar given the five different formulations.

Since the membrane strains are small, the use of the rigid bodies to impose the constraints on the rotation-free elements does not introduce a noticeable error. Details of the rigid body formulation are given in the Appendix. The rigid bodies accounted for approximately three percent to the total CPU time, indicating that they are an efficient approach for enforcing continuity between patches. The blended solution is less costly than the rotation-free solution primarily because of the reduced number of time steps. On an element per time step basis, the cost of the two formulations is nearly identical.

Table 2 summarizes the problem sizes and the CPU times for a single Xeon processor running the analyses in double precision. Comparing the computational costs of the isogeometric elements to the traditional elements, it is clear that the quadratic isogeometric elements are less costly primarily because of their larger stable time step size. The fully-integrated quadrilateral is the most expensive solution because it has a smaller time step size and four times as many integration points in the mesh as the quadratic NURBS. In this calculation, the Reissner-Mindlin formulation is the most accurate. However, we note that in implicit calculations there should be significant efficiency advantages in the blended formulation.

## 6 Summary and conclusions

There are significant potential savings in computational cost with rotation-free shells in implicit analysis. Unfortunately, actually achieving such savings requires dealing with required constraints at shell intersections, along folds and boundaries, and at the interfaces of NURBS patches. No general strategy exists for implementing the variety of situations that occur in practical engineering calculations. On the other hand, the classical Reissner-Mindlin element formulation does not incur any problems of this kind, at the price of twice as many degrees of freedom. In this work we have proposed a new shell formulation that blends thin-plate rotation-free theory with Reissner-Mindlin theory. By only selecting the Reissner-Mindlin formulation locally, where the constraints need to be enforced, we produce a formulation that yields the savings of the rotation-free formulation, combined with the robustness and generality of the Reissner-Mindlin formulation. We have tested the formulation on several problems and it has performed well in all cases. We have also identified some of the difficulties encountered implementing constraint equations in rotation-free calculations. Our computations have revealed that uniformly reduced integration quadratic NURBS elements are computationally efficient and, for the same level of accuracy, they rival the fastest known low-order one-point quadrature shell elements in speed. We believe that the isogeometric blended shell formulation has the potential to become the method of choice for a wide variety of practical engineering shell problems.

## Acknowledgments

D. Benson was supported by Honda R&D Americas, Inc. Award HRA 4500127023. Y. Bazilevs was supported through ARO Award 386 W911NF-11-1-0083. T.J.R. Hughes was supported by grants from the Office of Naval Research (N00014-08-1-0992) and SINTEF (UTA10-000374), with the University of Texas at Austin.

## A A Simple Rigid Body Dynamics Formulation for Explicit Calculations

Rigid bodies are a standard feature in LS-DYNA [14]. The algorithm [14,7] for updating their motion is summarized here.

The equations of motion for the center of mass for a rigid body are

$$m\ddot{\mathbf{x}}_{CM} = \mathbf{F} \quad \text{and} \quad \mathbf{J}\dot{\boldsymbol{\omega}} = \mathbf{M} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \quad (\text{A.1})$$

where  $m$  is the mass of the body,  $\mathbf{J}$  is the mass moment of inertia tensor,  $\mathbf{x}_{CM}$  are the coordinates of the center of mass,  $\boldsymbol{\omega}$  is the angular velocity, and  $\mathbf{F}$  and  $\mathbf{M}$  are the sums of the externally applied forces and moments, respectively.

The mass and the initial center of mass coordinates are

$$m = \sum_A m_A \quad (\text{A.2})$$

$$\mathbf{x}_{CM} = \frac{1}{m} \sum_A M_A \mathbf{x}_A \quad (\text{A.3})$$

where the summation is performed over all the control points (or nodes) in the rigid body. After determining the initial center of mass, the mass moment of inertia  $\mathbf{J}$  is calculated as

$$\mathbf{J} = \sum_A m_A \left( \|\mathbf{x}_A - \mathbf{x}_{CM}\|^2 \mathbf{I} - (\mathbf{x}_A - \mathbf{x}_{CM}) \otimes (\mathbf{x}_A - \mathbf{x}_{CM}) \right) + J_A \mathbf{I}. \quad (\text{A.4})$$

At  $t^n$ , the nodal forces  $\mathbf{F}_A^n$  and moments  $\mathbf{M}_A^n$  are summed to calculate the rigid body forces and moments,

$$\mathbf{F}^n = \sum_A \mathbf{F}_A^n \quad (\text{A.5})$$

$$\mathbf{M}^n = \sum_A (\mathbf{x}_A^n - \mathbf{x}_{CM}^n) \times \mathbf{F}_A^n + \mathbf{M}_A^n \quad (\text{A.6})$$

and they are substituted into Equation A.1.

Central difference time integration updates the rigid body velocity and the center of mass coordinates.

$$\dot{\mathbf{x}}_{CM}^{n+1/2} = \dot{\mathbf{x}}_{CM}^{n-1/2} + \Delta t^n \ddot{\mathbf{x}}_{CM}^n \quad (\text{A.7})$$

$$\dot{\boldsymbol{\omega}}^{n+1/2} = \dot{\boldsymbol{\omega}}^{n-1/2} + \Delta t^n \ddot{\boldsymbol{\omega}}^n \quad (\text{A.8})$$

$$\mathbf{x}_{CM}^{n+1} = \mathbf{x}_{CM}^n + \Delta t^{n+1/2} \dot{\mathbf{x}}_{CM}^{n+1/2} \quad (\text{A.9})$$

$$(\text{A.10})$$

The calculation of the incremental rotation vector

$$\Delta\boldsymbol{\theta}^{n+1} = \Delta t^{n+1/2} \dot{\boldsymbol{\omega}}^{n+1/2} \quad (\text{A.11})$$

is preferred to integrating a finite rotation measure for simplicity and robustness. Traditional finite rotation measures consisting of three angles have singular positions and quaternions require the imposition of a quadratic constraint [29]. The incremental rotation matrix  $\Delta\mathbf{R}$  is calculated from  $\Delta\boldsymbol{\theta}^{n+1}$  with the Hughes-Winget formula [18] and used to incrementally rotate the mass moment of inertia tensor from  $n$  to  $n + 1$ .

Finally, the control point velocity and coordinates are updated to  $n + 1/2$  and  $n + 1$ , respectively, in a manner guaranteed to exactly preserve the rigid body behavior of the system,

$$\dot{\mathbf{x}}_A^{n+1/2} = \dot{\mathbf{x}}_{CM}^{n+1/2} + \frac{1}{\Delta t^{n+1/2}} \Delta\mathbf{R} (\mathbf{x}_A^n - \mathbf{x}_{CM}^n) \quad (\text{A.12})$$

$$\mathbf{x}_A^{n+1} = \mathbf{x}_A^n + \Delta t^{n+1/2} \dot{\mathbf{x}}_A^{n+1/2}. \quad (\text{A.13})$$



## References

- [1] Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D. J. Benson. A computational procedure for prebending of wind turbine blades. *International Journal for Numerical Methods in Engineering*, 89:323–336, 2012.
- [2] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.
- [3] T. Belytschko, J. I. Lin, and C. S. Tsay. Explicit algorithms for the nonlinear dynamics of shells. *Computer Methods in Applied Mechanics and Engineering*, 42:225–251, 1984.
- [4] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 87(6):541–565, 2011.
- [5] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T. J. R. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):276–289, 2010.
- [6] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1367–1378, 2011.
- [7] D. J. Benson and J. O. Hallquist. A simple rigid body algorithm for structural dynamics programs. *International Journal for Numerical Methods in Engineering*, 22:723–749, 1986.
- [8] D. J. Benson and J. O. Hallquist. A single surface contact algorithm for the postbuckling analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 78:141–163, 1990.
- [9] M. Bischoff and E. Ramm. Shear deformable shell elements for large strains and rotations. *International Journal for Numerical Methods in Engineering*, 40:4427–4449, 1997.
- [10] Rui P.R. Cardoso and Jeong Whan Yoon. One point quadrature shell element with through-thickness stretch. *Computer Methods in Applied Mechanics and Engineering*, 194(911):1161 – 1199, 2005.
- [11] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal for Numerical Methods in Engineering*, 47:2039–2072, 2000.
- [12] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

- [13] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.
- [14] J. O. Hallquist. LS-DYNA theoretical manual. Technical report, Livermore Software Technology Corporation, 1998.
- [15] T. J. R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Dover, 2000.
- [16] T. J. R. Hughes and E. Carnoy. Nonlinear finite element shell formulation accounting for large membrane strains. *Computer Methods in Applied Mechanics and Engineering*, 39(1):69–82, 1983.
- [17] T. J. R. Hughes and W.K. Liu. Nonlinear finite element analysis of shells: Part I. Three-dimensional shells. *Computer Methods in Applied Mechanics and Engineering*, 26:331–362, 1981.
- [18] T. J. R. Hughes and J. Winget. Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis. *International Journal for Numerical Methods in Engineering*, 15(12):1862–1867, 1980.
- [19] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [20] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2403–2416, 2010.
- [21] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3902 – 3914, 2009.
- [22] E. Oñate and F. G. Flores. Advances in the formulation of the rotation-free basic shell triangle. *Computer Methods in Applied Mechanics and Engineering*, 194:2406–2443, 2005.
- [23] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*, 2nd ed. Springer-Verlag, New York, 1997.
- [24] D.F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.
- [25] D. Schillinger and E. Rank. An unfitted  $hp$ -adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(47-48):3358–3380, 2011.
- [26] S. Schoenfeld and D. J. Benson. Quickly convergent integration methods for plane stress plasticity. *Communications in Applied Numerical Methods*, 9:293–305, 1993.

- [27] T.W. Sederberg, G.T. Finnigan, X. Li, H. Lin, and H. Ipson. Watertight trimmed NURBS. *ACM Transactions on Graphics*, 27 (3), 2008.
- [28] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer, New York, 2000.
- [29] J. Wittenberg. *Dynamics of Multibody Systems*. Springer, 2007.